EXHIBIT G

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

SINGULAR COMPUTING LLC, Plaintiff, v. GOOGLE LLC, Defendant.	**Civil Action No. 1:19-cv-12551-FDS** **JURY TRIAL DEMANDED**

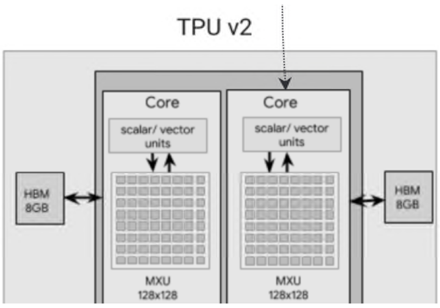
<u>AMENDED COMPLAINT FOR PATENT INFRINGEMENT</u>

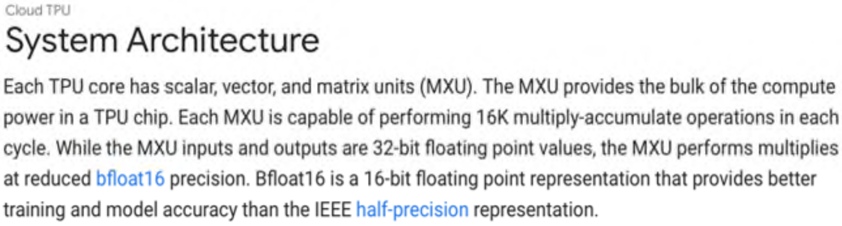
Plaintiff, Singular Computing LLC ("Singular"), for its amended complaint against Defendant, Google LLC, ("Google"), alleges as follows:

<u>THE PARTIES</u>

1. Singular is a Delaware limited liability company having its principal places of business at 10 Regent Street, Newton, MA 02465 and The Cambridge Innovation Center, 1 Broadway, Cambridge, MA 02142.

2. Google is a Delaware limited liability company and has regular and established places of business in this District, including a major office complex in Cambridge, Massachusetts with over 1,500 employees. Google may be served with process through its registered agent, Corporation Service Company, 84 State Street, Boston, MA 02109.

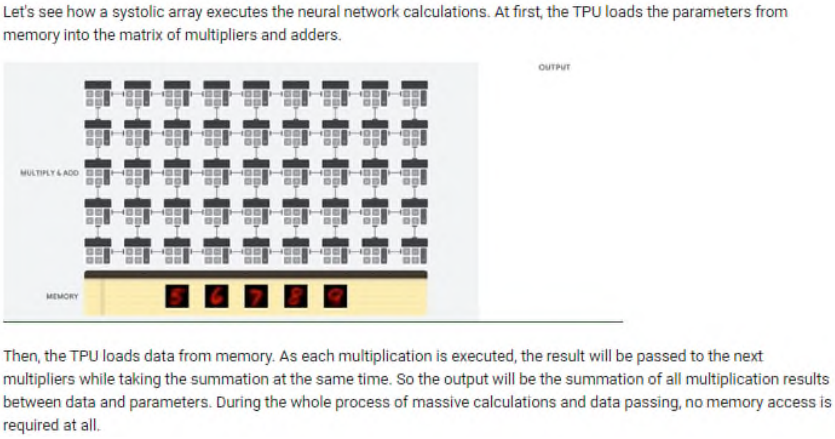




Cloud TPU

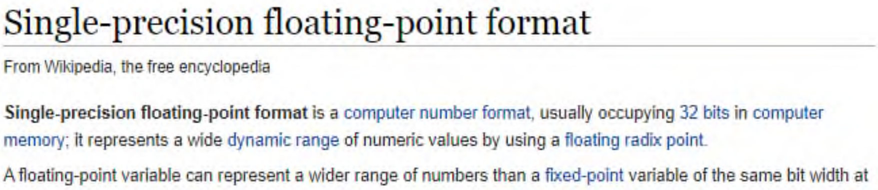
System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.



Let's see how a systolic array executes the neural network calculations. At first, the TPU loads the parameters from memory into the matrix of multipliers and adders.

Then, the TPU loads data from memory. As each multiplication is executed, the result will be passed to the next multipliers while taking the summation at the same time. So the output will be the summation of all multiplication results between data and parameters. During the whole process of massive calculations and data passing, no memory access is required at all.



Single-precision floating-point format

From Wikipedia, the free encyclopedia

Single-precision floating-point format is a computer number format, usually occupying 32 bits in computer memory; it represents a wide dynamic range of numeric values by using a floating radix point.

A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width at

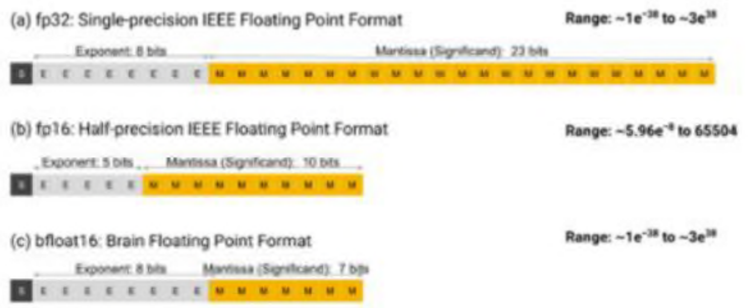
94. Each MXU Reduced Precision Multiply Cell is an "*LPHDR execution unit*." Specifically:

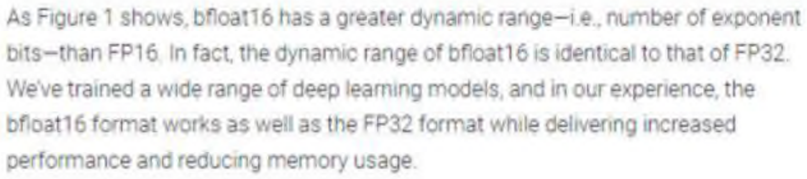
- For each MXU Reduced Precision Multiply Cell, "*the dynamic range of the possible valid inputs to the first operation is at least as wide as from 1/1,000,000 through 1,000,000.*" As shown above, each MXU Reduced Precision Multiply Cell performs a float32 multiplication operation at "reduced bfloat16 precision" on valid input signals representing numerical values having a float32 format. A float32 numerical value, whose format is shown below, has the following dynamic range:

Minimum: $2^{-126} \approx 1.175494351 \times 10^{-38}$

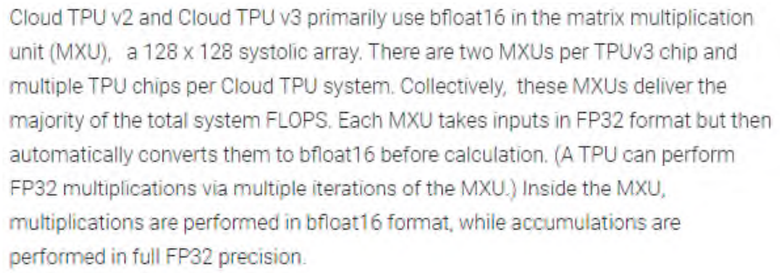
Maximum: $(2 - 2^{-23}) \times 2^{127} \approx 3.402823466 \times 10^{38}$

As published by Google:

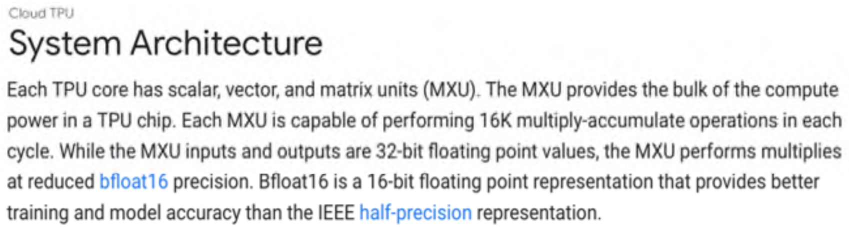

(a) fp32: Single-precision IEEE Floating Point Format
Range: ~1e^{-38} to ~3e^{38}
Exponent: 8 bits
Mantissa (Significand): 23 bits
S E E E E E E E E M M M M M M M M M M M M M M M M M M M M M M M
(b) fp16: Half-precision IEEE Floating Point Format
Range: ~5.96e^{-8} to 65504
Exponent: 5 bits
Mantissa (Significand): 10 bits
S E E E E E M M M M M M M M M M
(c) bfloat16: Brain Floating Point Format
Range: ~1e^{-38} to ~3e^{38}
Exponent: 8 bits
Mantissa (Significand): 7 bits
S E E E E E E E E M M M M M M M


As Figure 1 shows, bfloat16 has a greater dynamic range—i.e., number of exponent bits—than FP16. In fact, the dynamic range of bfloat16 is identical to that of FP32. We've trained a wide range of deep learning models, and in our experience, the bfloat16 format works as well as the FP32 format while delivering increased performance and reducing memory usage.

- For each MXU Reduced Precision Multiply Cell, "*for at least X=5% of the possible valid inputs to the first operation… the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input differs by at least Y=0.05% from the result of an exact mathematical calculation of the first operation on the numerical values of that same input*." Specifically, each TPUv2 and TPUv3 MXU Reduced Precision Multiply Cell performs a float32 multiplication operation but does so in Google's own words at "reduced bfloat16 precision." Each MXU Reduced Precision Multiply Cell takes the following steps: (i) receives as input two signals that each represent a float32 numerical value, (ii) converts each of the received float32 numerical values to a bfloat16 numerical value, (iii) multiplies the resulting pair of bfloat16 numerical values with each other, and (iv) adjusts the format of the result of the bfloat16 multiplication generated in step (iii), if needed, to produce an output signal that represents a float32 numerical value to be accumulated. As published by Google:



Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

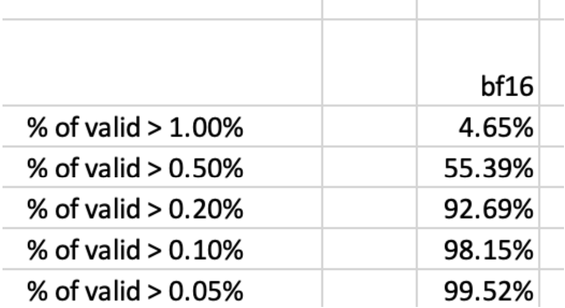


Cloud TPU

System Architecture

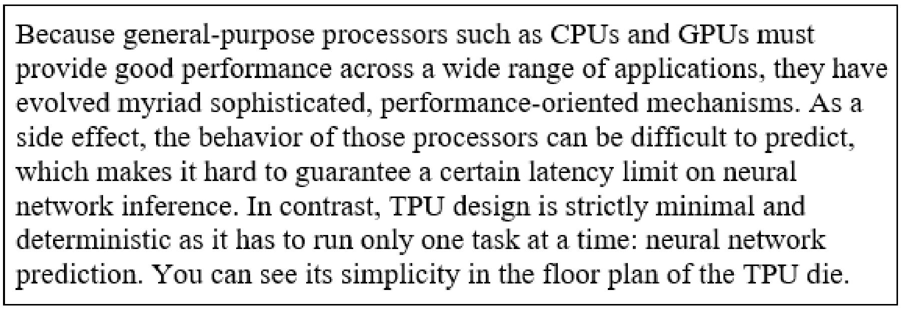
Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced bfloat16 precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE half-precision representation.

When the float32 numerical values produced by the TPU's float32 multiplication operation (which, as shown above, is performed at "reduced bfloat16 precision"), for a mathematically representative sample of all possible valid pairs of inputted float32 numerical values, are compared to the numerical values produced by the exact full precision multiplication operations for those same respective valid pairs of inputted float32 numerical values, the TPU's float32 numerical values differ, for at least 5% of those multiplied pairs, from the respective exact full precision values, by at least 0.05%. This is illustrated by the Singular test results shown below.



		bf16
% of valid > 1.00%		4.65%
% of valid > 0.50%		55.39%
% of valid > 0.20%		92.69%
% of valid > 0.10%		98.15%
% of valid > 0.05%		99.52%

- For each MXU Reduced Precision Multiply Cell, *"the statistical mean, over repeated execution of the first operation on each specific input from the at least X % of the possible valid inputs to the first operation, of the numerical values represented by the first output signal of the LPHDR unit executing the first operation on that input,"* will simply equal the numerical value represented by the output signal produced when the MXU Reduced Precision Multiply Cell (i.e., an LPHDR unit) executes an operation on input signals. Each MXU Reduced Precision Multiply Cell is part of a TPUv2 Device or a TPUv3 Device, which are deterministic in their designs (i.e., an operation repeatedly performed by a TPUv2 or a TPUv3 Device on a given set of inputs signals will always yield the same output signal). As published by Google:



Because general-purpose processors such as CPUs and GPUs must provide good performance across a wide range of applications, they have evolved myriad sophisticated, performance-oriented mechanisms. As a side effect, the behavior of those processors can be difficult to predict, which makes it hard to guarantee a certain latency limit on neural network inference. In contrast, TPU design is strictly minimal and deterministic as it has to run only one task at a time: neural network prediction. You can see its simplicity in the floor plan of the TPU die.

95. In each TPUv2 and each TPUv3 Device, *"the number of LPHDR execution units in the device exceeds by at least one hundred the non-negative integer number of execution units in the device adapted to execute at least the operation of multiplication on floating point numbers that are at least 32 bits wide."* As shown above, a TPUv2 Device has 8 MXUs, a